#5

JC869 U.S. PTO
09/637214
08/11/00

Europäisches Patentamt | European Patent Office | Office européen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

99115963.3

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE,
LA HAYE, LE 24/03/00

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 99115963.3
Demande n°:

Anmeldetag:
Date of filing: 11/08/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
International Business Machines Corporation
Armonk, NY 10504
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method and system for programming FPgas on PC-cards without additional hardware

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
G06F17/50

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

D E S C R I P T I O N

Method and System for Programming FPGAs on PC-cards without additional hardware

1. BACKGROUND OF THE INVENTION

1.1 FIELD OF THE INVENTION

The present invention relates to the area of programming field programmable gate arrays. In particular, the present invention relates to improve the usage of them during the use by end-users and during development of circuits implementing some new functionality on said field programmable arrays.

1.2 DESCRIPTION AND DISADVANTAGES OF PRIOR ART

In general, field programmable arrays, further referred to herein and abbreviated as FPGA are used to implement some computing functionality which is intended to run basic, hardware related functions used to control the basic functions of computer periphery, as, e.g., video screens, printers, network cards, etc.

Further, they are used in applications which require a vast amount of basic arithmetic computations as, e.g., multiply and add operations which have to be performed very quickly in graphic processing applications.

FPGA are used to implement computational functionality which is implemented in a large number (up to 1 million) of inter-connected circuits. Such vast cicuits are planned and realized with the help of a special hardware, as e.g., the hardware device of Byte BlasterMV which is connected to either the serial or the parallel port of the workstation used in turn as a development platform with dedicated software tools.

This is depicted in **fig. 2.** where a PCI card 18 is depicted schematically which is shown connected to an extra hardware 32

external to the PC. A logic 34 is provided for controlling and programming the FPGA 16 with the configuration data to be developed for the FPGA and feeding it with the configuration data necessary for the PCI card to be detected by the bus system on system start-up. Such prior art development environments require either additional hardware or Programmable Read Only Memory devices, further referred to herein as PROMs. The developed schema containing the new functionality is fed into the PROM which is placed onto the PC-card to be developed comprising the FPGA. After a subsequent POWER-On of the developer's workstation the PROM controls and performs the configuration of the FPGA. Then, the functionality of the FPGA can be tested during operation. If a further update of the schema is necessary, a new PRom has to be used, as the used one can not be rewritten.

Alternatively, Electrical Erasable PROMs, further referred to herein as EEPROMS are used instead of PROMs. They can be rewritten in case of a further update, but external hardware is necessary as well in order to control the write process into the EEPROM.

In the cases during development of FPGA functionality, or when an end-user shall exchange a PC-card 18 comprising said FPGA, due to any update or extension of functionality incorporated in his card, a manual access to the concerned card is thus necessary in order to replace the card by another one in the latter case or to replace at least the PROM used to program the FPGA in the case of a FPGA developer.

Basically, the same applies when instead of an update any new functionality shall be implemented on an FPGA.

Any manual access to PC-cards, however, causes additional work and bears the risk to damage other hardware connected in the casing of the computer, e.g., by statical charges brought to any of a plurality of locations sensitive thereto.

1.3 OBJECTS OF THE INVENTION

Therefore, an object of the present invention is to provide a method and circuit in order to improve the usage of FPGAs during the use by end-users and during development of circuits implementing some new functionality on said FPGAs.

2. SUMMARY AND ADVANTAGES OF THE INVENTION

These objects of the invention are achieved by the features stated in enclosed independent claims.

The inventional concept comprises method and system for programming updating hardware electronoc circuits without manually accessing the circuits.

According to the basic inventional concepts it is proposed to first physically connect a PROM or a EEPROM device on the concerned card. This device is intended to be accessed serially and to input a serial data stream into the FPGA, which is necessary when the FPGA shall be notified to the bus system of the computer, e.g., a PCI bus, and in order to be able to access the card via a conventional device driver. As mentioned above the FPGA is automatically configured by the PROM on the next Power-On of the computer which represents a requirement for the PCI-card to be detected properly by the BIOS of the computer.

Further, the inventional hardware circuit arrangement comprises an EEPROM device, and a FPGA device which is accessible via a computer bus system and a MUX element connected between said devices.

Further, in the circuit arrangement said PROM device is arranged for comprising control data for proper recognition of the FPGA by said bus system, and for comprising a logic usable for programming said EEPROM device with an EEPROM-FPGA interface like that of Joint Test Action Group (JTAG).According to the present invention the above mentioned MUX element can be controlled to

select either said PROM device or said EEPROM device or said FPGA device for reading data from said devices, in order to properly connect said FPGA to said bus system and to initialize a configuration of said FPGA with the contents comprised of said EEPROM.

The above described circuit arrangement can be used to perform various updates of the FPGA contents without accessing the card physically. This is achieved summarized as follows:

During a first sequence of steps the FPGA is used to program the EEPROM with the schema received from disk as mentioned above. Then, the MUX is switched to be able to read from the EEPROM and feed the developed schema programmed therein into the FPGA as it was intended originally. The PROM is just used to deliver the information to the FPGA which is necessary for the PC-card to be recognized by the BIOS on a first start-up. Thus, a characteristic feature of the present invention is that the FPGA device is configured in a double way, first in order to initialize the desired disk communication and then to be re-configured according to the EEPROM contents.

The method and circuit according to the present invention have the advantage, in relation to the method sketched out in the discussion of prior art technique that they allow to program FPGAs without accessing physically the card comprising the FPGA. Consequently, the average development time is lowered, cost are reduced.

Further, no extra hardware is required external to the computer.

Further, it is possible to construct a generic PC-card which can easily be re-programmed in order to perform an extended functionality compared to that before. Or, it is possible to implement a totally different functionality - if desired. Thus, the present invention increases flexibility of FPGA hardware as it is no more dedicated for a sole purpose only.

3. BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and is not limited by the shape of the figures of the accompanying drawings in which:

Fig. 1 is a schematic representation of a structural diagram showing the essential elements of the circuit according a preferred embodiment of the present invention,

Fig. 2 is a schematic representation of a structural diagram showing the essential elements of a circuit according to prior art,

Fig. 3 is a schematic representation of a block diagram showing the essential steps of the method according to a first and a second aspect of the present invention.

4. DESCRIPTION OF THE PREFERRED EMBODIMENT

With general reference to the figures and with special reference now to **fig. 1** the essential elements of the circuit according to the invention are described next below.

A PROM 10 and an EEPROM 12 is connected via a multiplexer element 14 with an FPGA 16 located on the generic PC-card to be generated according to the present invention. The PC-card is depicted schematically with broken lines with reference numeral 18. A number of 40 lines are provided at the right side of the FPGA as input/output lines in order to represent the connection to the PCI bus system of the computer.

From both, PROM 10 and EEPROM 12 a clock line and a data line are connected to respective entries of the MUX 14. A pair of clock and data line is output from the MUX to respective entries in the FPGA 16.

The MUX element 14 can be controlled via a line MUX CTL in order

to read data from the PROM , i.e., when said line is inactive 10, or from the EEPROM 12, when said line is switched active.

Further, there are provided four connections TCK, TDI, TMS and TDO between FPGA 16 and EEPROM 12 in order to program the EEPROM from the FPGA as discussed with reference to the prior art cited above.

Further, there is provided a signal line INIT_CONFIG from an output of the FPGA 16 to an input of the EPROM 12, the operation of which will be discussed later.

PROM 10 comprises all configuration data necessary to configure the FGPA in order to be recognized by the BIOS as a PCI-bus participating device on a start-up of the computer and to be accessed via a device driver. Further, it comprises all logic necessary to program the EEPROM 12 with the JTAG interface as discussed above.

With additional reference to **fig. 3** the essential steps of programming the EEPROM and the FPGA will be described next below.

After Power-On, step 110, the FPGA 16 is configured automatically via the PROM contents, step 120. The FPGA signals its presence after being prompted by the BIOS. The FPGA contains the PCI target device function and can thus communicate with a device driver.

In the case representing the first aspect of the inventional method, the EEPROM used to program the FPGA shall be programmed with a new update of the FPGA development schema mentioned above, what corresponds to the YES-branch in decision 130. Any prior art device driver reads the schema, i.e., the configuration data, e.g., from a disk, step 140, to where it was written by the schema development tool and programs these data into the EEPROM 12 via the dedicated function implemented in the FPGA which was mentioned above, step 150.

In particular, like in prior art, the dedicated JTAG signals Clock (TCK), Data Input (TDI), Mode selection (TMS) and Data Output (TDO) are used as depicted in fig. 1. Thus, the EEPROM's programming is completed.

Then, the configuration process for the FPGA 16 with the contents of the EEPROM 12 is triggered by activating the so-called Card_INIT function via the signal line INIT_CONFIG. Said triggering step is controlled by a function implemented in the FPGA.

In particular, the MUX 14 is switched in step 160 via the line MUX CTL so that the lines CLK and DATA of the EEPROM 12 are fed into the FPGA 16. When the line INIT_CONFIG is actually activated, step 170, the FPGA will be configured with the contents of the EEPROM, step 180. Thus, the FPGA programming is completed, step 190. It can be repeated with a new updated version of the schema by simply repeating the steps described just described above.

According to the second aspect of the inventional method the EEPROM 12 is already re-programmed. Thus, such cases are covered in which the development of a new FPGA configuration schema is completed.

In this case the method as depicted in fig. 3 continues after the start-up procedure, step 110 with the PROM involved, step 120, leaves the decision 130 via the NO-branch and continues with step 160 as described above.

In the foregoing specification the invention has been described with reference to a specific exemplary embodiment thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are accordingly to be regarded as illustrative rather than in a restrictive sense.

It should be noted that the inventional concepts are independent from the bus system in use with the PC hosting the FPGA schema development environment.

C L A I M S

1. A method for programming field programmable gate arrays (FPGA) (16) with configuration data according to a schema developed by a developing tool on a computer device, the method being characterized by the steps of

reading (140) said schema by a device driver from a storage device of said computer device,

programming (150) said schema by aid of a dedicated function implemented in said FPGA (16) into an EEPROM (12) connected with said FPGA (16) via a MUX element (14),

switching (160) the MUX element (14) in order to be able to read from said EEPROM (12) into said FPGA (16), and

triggering (170) the configuration of said FPGA (16) by feeding said schema from said EEPROM (12) to said FPGA (16).

2. A method for using FPGA (16) with configuration data stored in a EEPROM (12) connected to said FPGA (16) via a MUX element (14), the method being characterized by the steps of

controlling said MUX element (14) in order to be able to read from said EEPROM (12) into said FPGA (16), and

triggering the configuration of said FPGA (16) by feeding said schema from said EEPROM (12) to said FPGA (16).

3. A hardware circuit arrangemént comprising a PROM device (10), an EEPROM device (12), a FPGA device (16) accessible via a computer bus system and a MUX element (14) connected between said devices, the circuit arrangement being characterized by

said PROM device (10) being arranged for comprising control

data for proper recognition of said FPGA (16) by said bus system, and a logic usable for programming said EEPROM device (12) with an EEPROM-FPGA interface,

said MUX element (14) being controllable to select either said PROM device (10) or said EEPROM device (12) or said FPGA device (16) for reading data from said devices, in order to properly connect said FPGA (16) to said bus system and to initialize a configuration of said FPGA (16) with contents comprised of said EEPROM (12).

4. PC-card (18) detectable by a PC system bus and comprising a circuit arrangement according to claim 3.

LIST OF REFERENCE SIGNS

10	PROM
12	EEPROM
14	MUX
16	FPGA
18	PC-card
32	external hardware
34	logic
110 to 190	steps, decisions of the inventional method

A B S T R A C T

The inventive concept comprises a method and a circuit arrangement for programming or updating hardware electronic circuits without manually accessing the circuits. The inventive hardware circuit arrangement comprises an EEPROM device (12), a FPGA device (16) which is accessible via a computer bus system and a MUX element (14) connected between said devices.

Further, in the circuit arrangement a PROM device (10) is arranged for comprising control data for proper recognition of the FPGA by said bus system, and for comprising a logic usable for programming said EEPROM device with an EEPROM-FPGA interface like that of Joint Test Action Group (JTAG).The above mentioned MUX element can be controlled to select either said PROM device or said EEPROM device or said FPGA device for reading data from said devices, in order to properly connect said FPGA to said bus system and to initialize a configuration of said FPGA with the contents comprised of said EEPROM.

During a first sequence of steps the FPGA is used to program the EEPROM with the schema received from disk as mentioned above. Then, the MUX is switched to be able to read from the EEPROM and feed the developed schema programmed therein into the FPGA as it was intended originally. The PROM is just used to deliver the information to the FPGA which is necesary for the PC-card to be recognized by the BIOS on a first start-up. (Fig. 1)

[drawings)

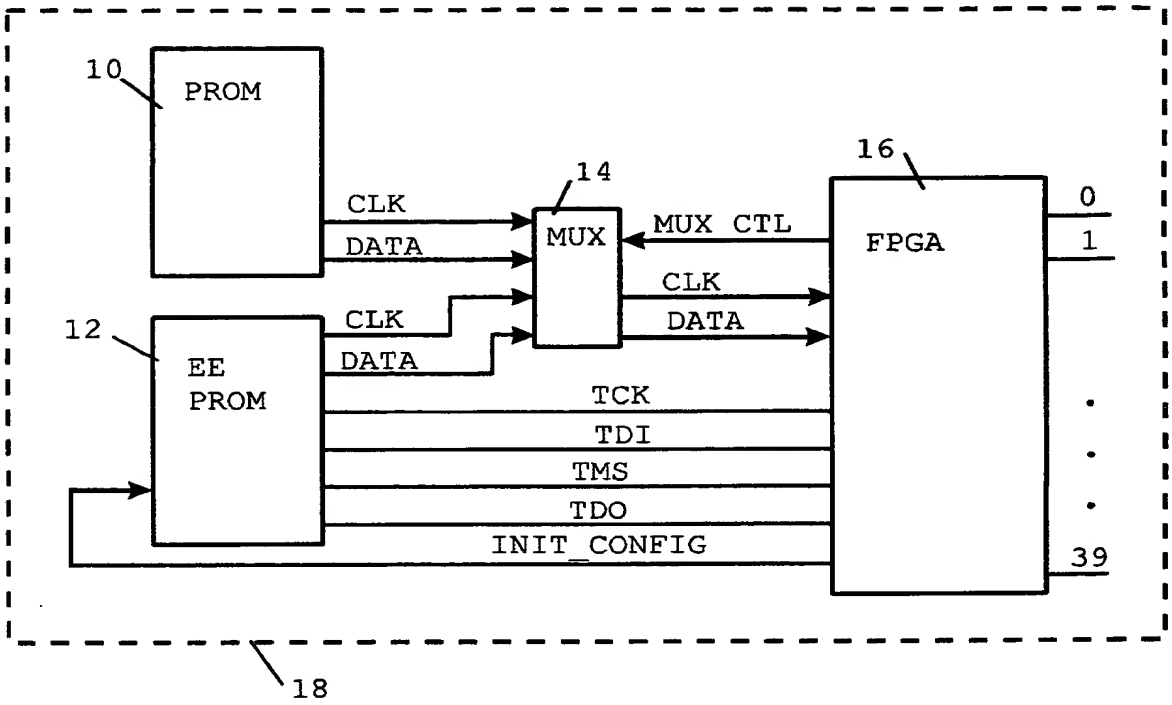

FIG. 1

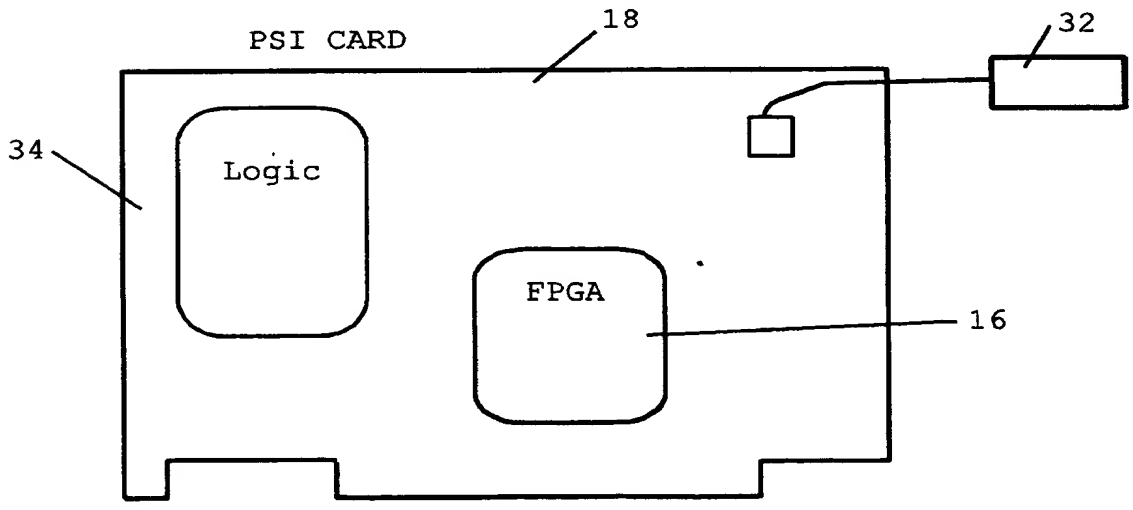

FIG. 2

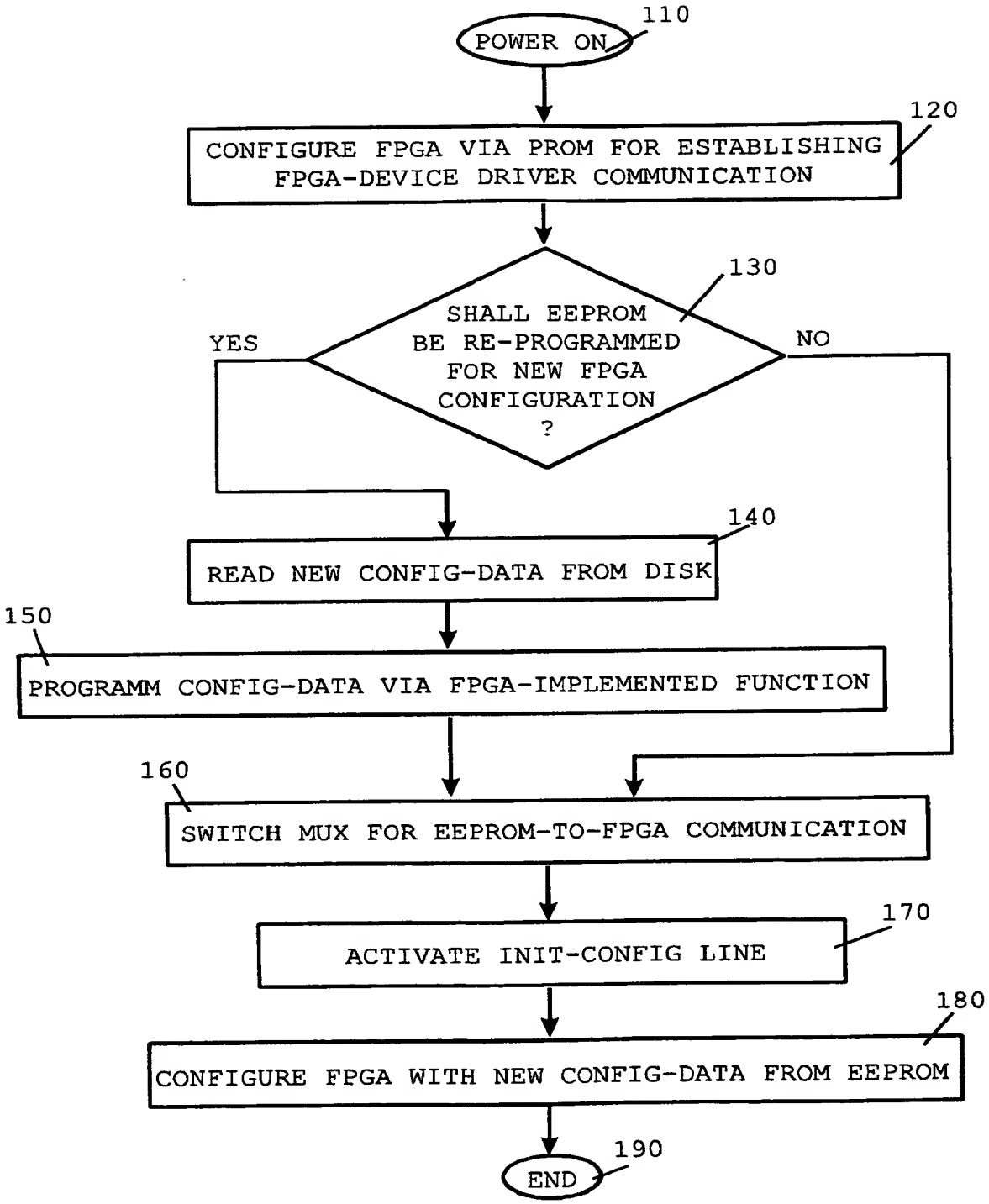
110
POWER ON
120
CONFIGURE FPGA VIA PROM FOR ESTABLISHING FPGA-DEVICE DRIVER COMMUNICATION
130
SHALL EEPROM BE RE-PROGRAMMED FOR NEW FPGA CONFIGURATION ?
YES
NO
140
READ NEW CONFIG-DATA FROM DISK
150
PROGRAMM CONFIG-DATA VIA FPGA-IMPLEMENTED FUNCTION
160
SWITCH MUX FOR EEPROM-TO-FPGA COMMUNICATION
170
ACTIVATE INIT-CONFIG LINE
180
CONFIGURE FPGA WITH NEW CONFIG-DATA FROM EEPROM
190
END

FIG. 3